JAVA.UTIL.BITSET CLASS

Introduction

The **java.util.BitSet** class implements a vector of bits that grows as needed.Following are the important points about BitSet:

- A BitSet is not safe for multithreaded use without external synchronization.
- All bits in the set initially have the value false.
- Passing a null parameter to any of the methods in a BitSet will result in a NullPointerException.

Class declaration

Following is the declaration for **java.util.BitSet** class:

```
public class BitSet
   extends Object
      implements Cloneable, Serializable
```

Class constructors

S.N.	Constructor & Description
1	**BitSet()** This constructor creates a new bit set.
2	**BitSet(int nbits)** This constructor creates a bit set whose initial size is large enough to explicitly represent bits with indices in the range 0 through nbits-1.

Class methods

S.N.	Method & Description
1	void and(BitSet set) This method performs a logical AND of this target bit set with the argument bit set.
2	void andNot(BitSet set) This method clears all of the bits in this BitSet whose corresponding bit is set in the specified BitSet.
3	int cardinality() This method returns the number of bits set to true in this BitSet.
4	void clear() This method sets all of the bits in this BitSet to false.
5	void clear(int bitIndex)

	This method sets the bit specified by the index to false.
6	void clear(int fromIndex, int toIndex) This method sets the bits from the specified fromIndex (inclusive) to the specified toIndex (exclusive) to false.
7	Object clone() This method clones this BitSet and produces a new BitSet that is equal to it.
8	boolean equals(Object obj) This method compares this object against the specified object.
9	void flip(int bitIndex) This method sets the bit at the specified index to the complement of its current value.
10	void flip(int fromIndex, int toIndex) This method sets each bit from the specified fromIndex (inclusive) to the specified toIndex (exclusive) to the complement of its current value.
11	boolean get(int bitIndex) This method returns the value of the bit with the specified index.
12	BitSet get(int fromIndex, int toIndex) This method returns a new BitSet composed of bits from this BitSet from fromIndex (inclusive) to toIndex (exclusive).
13	int hashCode() This method returns the value of the bit with the specified index.
14	boolean intersects(BitSet set) This method returns true if the specified BitSet has any bits set to true that are also set to true in this BitSet.
15	boolean isEmpty() This method returns true if this BitSet contains no bits that are set to true.
16	int length() This method returns the "logical size" of this BitSet: the index of the highest set bit in the BitSet plus one.
17	int nextClearBit(int fromIndex) This method returns the index of the first bit that is set to false that occurs on or after the specified starting index.
18	int nextSetBit(int fromIndex) This method returns the index of the first bit that is set to true that occurs on or after the specified starting index.
19	void or(BitSet set) This method performs a logical OR of this bit set with the bit set argument.
20	void set(int bitIndex) This method sets the bit at the specified index to true.
21	void set(int bitIndex, boolean value) This method sets the bit at the specified index to the specified value.
22	void set(int fromIndex, int toIndex) This method sets the bits from the specified fromIndex (inclusive) to the specified toIndex (exclusive) to

	true.
23	void set(int fromIndex, int toIndex, boolean value) This method sets the bits from the specified fromIndex (inclusive) to the specified toIndex (exclusive) to the specified value.
24	int size() This method returns the number of bits of space actually in use by this BitSet to represent bit values.
25	String toString() This method returns a string representation of this bit set.
26	void xor(BitSet set) This method performs a logical XOR of this bit set with the bit set argument.

Methods inherited

This class inherits methods from the following classes:

- java.util.Object